

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,
please do not report the images to the
Image Problem Mailbox.**

THIS PAGE BLANK (USPTO)

PCTWORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification ⁶ : H04Q 3/00		(11) International Publication Number: WO 99/56475
A1		(43) International Publication Date: 4 November 1999 (04.11.99)
(21) International Application Number: PCT/SE99/00664		(81) Designated States: AE, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, UZ, VN, YU, ZA, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).
(22) International Filing Date: 23 April 1999 (23.04.99)		
(30) Priority Data: 09/069,168 29 April 1998 (29.04.98) US 09/143,619 28 August 1998 (28.08.98) US		
(71) Applicant: TELEFONAKTIEBOLAGET LM ERICSSON (publ) [SE/SE]; S-126 25 Stockholm (SE).		
(72) Inventors: JOHANSSON, Staffan, Engelbert; Duvhöksgränd 15, S-974 55 Luleå (SE). RYNBÄCK, Patrik, Johan, Erik; Professorsvägen 143 A, S-977 51 Luleå (SE). ANDERSSON, Christoffer; Kårhusvägen 2:155, S-977 54 Luleå (SE). JÖNSSON, Nils, Tore, Erik; Markvägen 25, S-974 51 Luleå (SE). JOHANSSON, Peter; Västra Varvsgatan 14 B, S-972 36 Luleå (SE).		
(74) Agent: ERICSSON RADIO SYSTEMS AB; Common Patent Dept., S-164 80 Stockholm (SE).		

Published

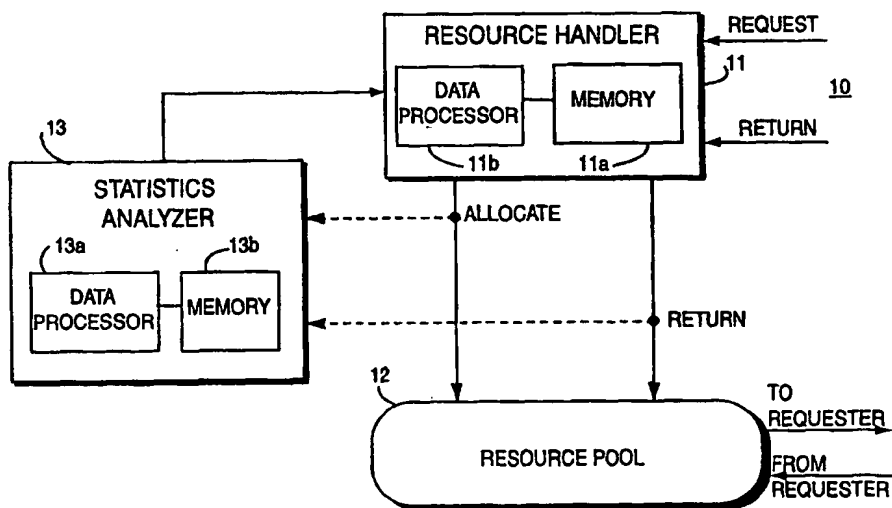
With international search report.

(54) Title: RESOURCE ALLOCATION

(57) Abstract

A pool of resource units is configured using a structure having plural levels including first level and second level resource blocks. Each first level resource block corresponds to an individual resource unit, where an individual resource unit is the smallest resource that can be allocated from the resource pool. The first level resource blocks are provided in groups. Each second level resource block corresponds to one of the groups of the first level resource blocks. The second level resource blocks are also provided in groups forming a next higher level of the resource pool. If a request

is made for a first level resource block, an available first level resource block is allocated. If a first level resource block is not available, then an available first level resource block is created by converting a second level resource block into a corresponding group of first level resource blocks. One of the converted first level resource blocks is then allocated to the request. Resource blocks are restored to the resource unit pool structure first to any incomplete first level resource block grouping. If that returned block completes the grouping, a new second level resource block is made available. The resource pool may be reconfigured as needed to ensure an efficient configuration. First and second pools of resource units are configured using a structure having plural levels including first and second level resource units. A relationship is established between one of the two pools of resource units. A request is made for a resource that requires a resource unit from both of the first and second resource unit pools. A single allocation procedure is employed to allocate resource units from both pools. If one of the resource units is available in one pool, the relationship ensures that the required corresponding resource unit in another pool is also available.



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece	ML	Mali	TR	Turkey
BG	Bulgaria	HU	Hungary	MN	Mongolia	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MR	Mauritania	UA	Ukraine
BR	Brazil	IL	Israel	MW	Malawi	UG	Uganda
BY	Belarus	IS	Iceland	MX	Mexico	US	United States of America
CA	Canada	IT	Italy	NE	Niger	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NL	Netherlands	VN	Viet Nam
CG	Congo	KE	Kenya	NO	Norway	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NZ	New Zealand	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	PL	Poland		
CM	Cameroon	KR	Republic of Korea	PT	Portugal		
CN	China	KZ	Kazakistan	RO	Romania		
CU	Cuba	LC	Saint Lucia	RU	Russian Federation		
CZ	Czech Republic	LI	Liechtenstein	SD	Sudan		
DE	Germany	LK	Sri Lanka	SE	Sweden		
DK	Denmark	LR	Liberia	SG	Singapore		
EE	Estonia						

RESOURCE ALLOCATION

RELATED APPLICATION

This is a continuation-in-part application of commonly assigned U.S. Patent
5 Application Serial No. 09/069,168, entitled "Method and Apparatus for Allocating
Processing Resources," filed on April 29, 1998.

FIELD OF THE INVENTION

The present invention relates to allocating resources in general, and more
particularly, to allocating various data, signal, and/or communications processing
10 resources.

BACKGROUND AND SUMMARY OF THE INVENTION

In certain environments, data, signal, and communication processing
resources may be dedicated for use by particular hardware elements, software elements
(e.g., application programs), and human operators, each of which may be generally viewed
15 as a "resource user." While such resource dedication is convenient and also ensures that
the necessary or otherwise appropriate resources are available whenever the resource user
requests them, this one-to-one, dedicated relationship between a resource and a resource
user is inefficient. This inefficiency is primarily due to the fact that dedicated resources are
most often underutilized since the resource user typically uses its dedicated resource less
20 than 100% of the time.

The present invention employs a more efficient resource allocation
approach in which resources are pooled and allocated on an "as needed" basis. In other
words, in response to a resource request, a resource (if available) is "withdrawn" from a
pool of resources and provided to the requesting resource user. When the resource user is

finished using that resource, it is returned to the resource pool. Efficiency is increased because (1) the number of resource users may be considerably larger than the number of resources in the pool and (2) resources are removed from the pool only when they are actually used.

5 In order for a pooled resource system to work smoothly, the allocation and return of resources from and to the resource pool needs to be managed in an effective and efficient manner. Efficient management is more challenging when the pool of resources as well as the resource requests are not all the same. For instance, one resource request may be for a relatively small resource while another resource request might be for a relatively
10 large resource. Moreover, resource requests may also differ in parameters other than size.

 The present invention resolves these issues in part by imposing a structure on the resource pool including specific rules regarding how resources are to be allocated from the pool. Those rules may vary depending on the end application that requests and uses the resources. One example (but non-limiting) structure is the resource hierarchy
15 shown in Fig. 1. The hierarchy includes a plurality of levels or tiers with each level having plural nodes representing a resource block at that level. The data structure is organized in such a way that it "mirrors" the physical resources in terms of how those physical resources are linked or grouped. The physical resources are configured, reconfigured, allocated, de-allocated, organized, and re-organized using the data structure.

20 In general, a number of lower level resource units are grouped to form a corresponding, higher level resource unit. Level 1 resource blocks correspond as shown in Fig. 1 to one data processing resource unit, where a resource unit may comprise hardware and/or software. Each level 2 resource block corresponds to a group (in this example two) of consecutive level 1 resource blocks. Similarly, each level 3 resource block of the
25 resource hierarchy corresponds to a group of consecutive level 2 resource blocks.

 Thus, if a resource request requires four data processing resource units, a single level 3 resource block would be allocated to that request. A level 3 resource block

corresponds, in this example, to two level 2 resource blocks which in turn corresponds to four level 1 resource blocks equivalent to the four data processing resource units.

While resources may be allocated from and returned to such a hierarchical structure in random fashion, such an approach may ultimately lead to inefficiencies. In particular, the initial, ordered structure of the resource hierarchy may become fragmented after several resource block allocations and returns. Unfortunately, because of the fragmentation, resource blocks at certain levels cannot be fully utilized. For example, Fig. 2 shows the resource hierarchy of Fig. 1 with a fragmented resource allocation configuration. Circled resource blocks correspond to a currently allocated resource block, and a square enclosing a resource block represents a currently-available resource block. Because level 1 resource block 14 is currently allocated, level 2 resource block 16, logically linked to both level 1 resource blocks 14 and 15, cannot be allocated. A "domino effect" occurs – since level 3 resource block 17 is linked to resource blocks 16 and 18, level 3 resource block 17 cannot be currently allocated even if level 2 resource block 18 were currently available.

On the other hand, if the currently-allocated resource block 19 had been allocated from currently-available resource block 15, the level resource block 18 would have been available for allocation. Similarly, if currently-allocated, level 1 resource block 21 was not allocated, and instead the currently-available level 1 resource block 23 was assigned to the resource request, the currently-unavailable level 2 resource block 21 would be available. The level 2 resource block 21 could then be combined with the level 2 resource block 25 to make available the level 3 resource block 26. Accordingly, as a result of fragmented/inefficient resource allocation at lower levels in the hierarchy, it may be difficult to provide a higher level resource block to a resource request even though there are sufficient, available resources in the resource pool to accommodate the higher level resource request.

In a pooled resource system where resources are allocated according to a particular strategy, different types or groups of pooled resources will likely have different resource allocation structures or algorithms. If more than one type of resource is required

to satisfy individual service requests, coordinating the allocation of both types of resources using two or more resource structures and/or algorithms becomes complicated and inefficient.

Consider the following example in a Code Division Multiple Access (CDMA) communications system. In order for a user to transmit information, the user must be allocated a spreading code (a software resource) corresponding to a particular channel capacity. Certain spreading codes have a smaller capacity. Other spreading codes in a spreading code resource hierarchy or tree may correspond to plural ones of smaller spreading code resources. As a result, a higher capacity, spreading code parent can only be allocated if its corresponding lower capacity, spreading code children are all available. If one of those spreading code children is already allocated, the higher capacity channel spreading code parent cannot be allocated.

In addition to a software spreading code resource, a user must be allocated the necessary hardware resource to physically perform the communication. An example of such hardware includes Digital Signal Processing (DSP) and/or radio transceiver circuitry. For example, a lowest capacity channel might use only one DSP and/or transceiver hardware unit. On the other hand, a higher capacity channel may require more than one DSP and/or transceiver hardware unit. Similarly, the higher capacity parent can only be allocated if all its supporting children are available. The fact that two different resource hierarchies, (i.e., software and hardware resources) in the above example, must be navigated makes the provision of an appropriate capacity spreading code with appropriate capacity hardware resources particularly cumbersome and inefficient. Not only are two resource allocation strategies required for each pool of resources, there is no guarantee that just because a suitable resource is available from one pool that a corresponding and necessary resource from the other pool is also available.

It is an object of the present invention to overcome inefficiency by providing a resource pool structure including a data structure reflecting the free resources in the pool and that permits organized and efficient but still flexible allocation and de-allocation of resources.

It is a further object of the present invention to ensure that resource units are evenly utilized in a resource pool.

It is another object of the present invention to selectively re-allocate resources among the resource users to free other resources that can be used to form higher level resource blocks in order to meet a particular resource demand or to simply make
5 available higher level resource blocks.

It is another object of the present invention to determine and implement a resource pool structure data structure, mirroring the free resources in the resource pool, that best suits current resource demands.

10 It is yet another object of the present invention to ensure that different types of resource units, e.g., hardware resources and software resources, are cooperatively allocated in response to resource requests in an efficient and effective manner.

The present invention configures a pool of resource units using a logical structure that includes first level and second level resource blocks. The terms "configure
15 resources" or "configuration of resources" refer to a data structure which is organized to reflect or represent actual or physical resources. However, the actual resources themselves are not necessarily configured in a resource hierarchy like that shown in Fig. 1. Each first level resource block corresponds to an individual resource unit, where an individual resource unit is the smallest resource that a resource handler allocates from the resource
20 pool. The first level resource blocks are provided in groups that form a lowest level of the resource pool structure. Each second level resource block corresponds to one of the groups of the first level resource blocks. For example, two first level resource blocks form a first level resource block group logically linked to a second level resource block. The second level resource blocks are provided in groups and form a next higher level of the
25 resource pool structure.

Resources are allocated from the resource unit pool in an efficient manner. For example, if a request is made for a first level resource block, an available first level resource block is allocated from a partially-allocated, first level resource group where one

or more first level resource blocks are currently already allocated. If a first resource block is not available, then an available first level resource block is created. In this case, a second level resource block is converted into a corresponding group of first level resource blocks, and one of the converted, first level resource blocks is then allocated to the request. A
5 similar procedure is used at a next higher level, i.e., at third level resource blocks and groupings. In other words, if a first resource block is not available at the first level and cannot be obtained by conversion of a resource block at the second level, a third level resource block is converted into a corresponding group of second level resource blocks; a converted second level resource block is converted into a corresponding group of first
10 level resource blocks; and one of the converted first level resource blocks is allocated.

Resource blocks are also restored to the resource unit pool structure efficiently. For example, a returned first level resource block is restored to a partially-allocated, first level resource block grouping. If that returned block completes that grouping, a new, second level resource block is made available. In one preferred
15 embodiment, when a resource block is restored, the physical resource is restored to the same position in the resource array as from where it was originally allocated. However, the corresponding resource block in the data structure is placed last in a list of free resource blocks. This ensures that each of the physical resources in the pool is used about the same amount over time.

20 The resource pool is managed by a resource handler which stores in memory a data structure that mirrors the current status (free or occupied) of the resources in the resource pool. The resource pool structure represents and maintains the logical relation between resource units/blocks. For example, the data structure may include multiple lists, each list corresponding to a resource block level and each record/entry in the list
25 representing one free resource unit. The list does not include allocated resource units.

To properly manage the resource unit pool or to otherwise meet a particular a resource request/demand, a resource handler may re-allocate resource units by moving a resource user from one resource block to another at that same level in order to create additional resource blocks at the next higher level. The resource handler may also receive

traffic statistics about the number of requests for different level resource blocks collected by a statistics analyzer and determine which level in the resource unit pool structure receives the greatest number of requests, i.e., the most popular level. In an example communication system, assume the system capacity is eight voice calls or two data calls.

5 Most of the calls in the system during busy hours are voice calls. During busy hours, if the whole system capacity is allocated for data calls, no voice calls can be made resulting in an insufficient subscriber service level in the area. By restricting the use of the transmission capacity, at least a minimum level of service for data calls and voice calls can be assured. The resource handler then re-allocates resource units in the current resource
10 pool configuration to create additional resource blocks at the most popular level.

The present invention also cooperatively allocates different types of resource units in response to a resource request in an efficient and effective manner. An example of different types of resource units that may be cooperatively allocated include software resources and hardware resource units. A first data structure may be configured to
15 correspond to a first pool of a first type of resource unit, e.g., software resource units. A second data structure may be configured to correspond to a second pool of a second type of resource unit, e.g., hardware resource units. A relationship is established between resource units in the first and second resource unit pools so that if one of the first resource units is available for allocation, a corresponding second resource unit is also assured
20 available. Then when a request is received for a resource, one or more available resource units from both the first and second resource pools may be allocated using the established relationship.

The established relationship between two resource pools may be implemented, for example, using a mapping mechanism such as a matrix, where each row
25 in the matrix represents different resource units of the first type and each column represents different resource units of the second type. Selection of a row corresponding to a resource unit of a first type automatically selects a corresponding resource unit of the second type associated with an activated column in the matrix. For a one-to-one type example relationship, the matrix is configured so that each row actively intersects with only

one of the columns. The mapping relationship can be reconfigured by changing which row-column intersections are active.

Using the established relationship, only one resource selection algorithm is employed which simplifies resource allocation procedures involving more than one pool of resource units. Thus, the use of two or more separate resource selection algorithms is avoided. The need for "synchronizing" separate resource algorithms is also avoided. In other words, without the present invention, failure to synchronize plural resource selection algorithms leads to situations (not encountered in the present invention) where an available resource unit in one pool is allocated without a corresponding and necessary resource unit from another pool being available, e.g., as a result of different fragmentations resulting from previously allocated resources.

The present invention has particularly advantageous application to communications systems, and in particular, to radio communications systems. A preferred example embodiment of the present invention is disclosed in the context of a wide band, code division multiple access (WCDMA), spread spectrum, radio communications system. In the context of a W-CDMA radio communications system, the first pool of resource units may correspond to W-CDMA spreading codes (software type resource units), and a second pool of resource units may correspond to data processing and transceiver units (hardware resource units). Of course, as evident from the above description, the present invention has general applicability and is not limited to communications environments. Various features and advantages of the present invention, some of which have been described above, will be disclosed in further detail in conjunction with the drawings and detailed description.

BRIEF DESCRIPTION OF THE DRAWINGS

These and other objects, features, and advantages of the present invention will now be described in conjunction with the drawings in which:

Fig. 1 is a diagram showing a resource unit hierarchy which is one example structure that may be imposed upon a resource pool;

Fig. 2 is a diagram showing a fragmented resource allocation using the resource unit hierarchy from Fig. 1;

5 Fig. 3 is a resource management system in accordance with a general embodiment of the present invention;

Fig. 4 is an example of an efficient resource allocation for the resource unit hierarchy of Fig. 1 in accordance with one example application of the present invention;

10 Fig. 5 is a flowchart diagram illustrating one set of example procedures for efficiently allocating resources in accordance with one aspect of the present invention;

Fig. 6 is a flowchart diagram illustrating one set of example procedures for efficiently returning resources in accordance with one aspect of the present invention;

Fig. 7 is a flowchart diagram illustrating one set of example procedures for reconfiguring a resource structure in accordance with one aspect of the present invention;

15 Fig. 8 is a function block diagram illustrating a radio communications system in which the present invention may be advantageously applied is a function block diagram illustrating a set of procedures for efficiently returning resources to the resource pool in accordance with one preferred but nevertheless example embodiment of the present invention;

20 Fig. 9 is a function block diagram illustrating in further detail an implementation of the present invention in a base station in a wideband code division multiple access radio communications system in accordance with one preferred but example embodiment of the present invention;

Fig. 10 is a resource management system in accordance with another embodiment of the present invention that draws upon resource units from different resource pools;

Fig. 11 is a flowchart diagram illustrating an example set of procedures for mapping resource units of different types in accordance with an example embodiment of the present invention; and

Fig. 12 is a diagram illustrating example hierarchical type data structures for first and second resource pools which are related together by way of a reconfigurable resource mapping matrix.

DETAILED DESCRIPTION OF THE DRAWINGS

In the following description, for purposes of explanation and not limitation, specific details are set forth, such as particular embodiments, data flows, techniques, etc. in order to provide a thorough understanding of the present invention. However, it will be apparent to one skilled in the art that the present invention may be practiced in other embodiments that depart from these specific details. For example, while a preferred example embodiment of the present invention is described in the context of an example of a WCDMA radio network, those skilled in the art will appreciate that the present invention has wide application to communications and data processing systems. In other instances, detailed descriptions of well-known methods, interfaces, devices, and signaling techniques are omitted so as not to obscure the description of the present invention with unnecessary detail.

Referring to Fig. 3, a general embodiment of the present invention as a resource management system 10 configures a data structure mirroring the free resources in a pool of resources 12 using a hierarchical or other layered structure. One example hierarchical structure is shown in Fig. 1 including resource blocks at various levels one, two, three, ... N. Resource blocks at each level are configured into groups, each group linked to one resource block at the next higher level. Thus, in the example in Fig. 1, two

level 1 resource blocks form a level 1 resource block group. A single level 2 resource block is linked to one of the level 1 resource block groupings. A single level 3 resource block is linked to a level 2 resource block grouping corresponding in the example shown in Fig. 1 to two consecutive level 2 resource blocks. Of course, groupings and/or structures other than that shown in Fig. 1 may be employed.

The resource management system 10 shown in Fig. 3 also includes a resource handler 11 and a statistics analyzer 13 in order to efficiently and otherwise optimally manage the allocation and return of resources from resource pool 12. The resource handler 11 includes a memory 11a and a data processor 11b. The statistics analyzer 13 also includes a data processor 13a and suitable memory 13b. As explained above, the statistics analyzer 13 monitors one or both of the allocations and returns of resources from/to resource pool 12. The statistics analyzer provides this information in an appropriate format to resource handler 11 which may, depending upon an analysis of the statistics, reorganize or otherwise reconfigure the resource pool hierarchy. Although the resource handler 11 and statistics analyzer 13 are shown as separate elements, each having its own data processor and memory, those skilled in the art will appreciate that they may be implemented using common data processing circuitry and memory.

In general, (though not always), the resource handler 11 attempts to allocate resource blocks to the extent possible so that higher level resource blocks are preserved for allocation. Consequently, the resource handler 11 first tries to allocate resource blocks from a grouping in which another resource block has already been allocated. Only when all of the resource blocks in a partially-allocated grouping are all currently allocated does the resource handler 11 draw upon another resource block group at that same level. When there are no available groups at the requested resource level, the resource handler 11 converts an available resource block from a next higher level (again from a group that is currently partially-allocated if possible) into a group at the requested resource block level. Then one of the converted resource blocks is allocated to the resource request. If there are no available resource blocks for conversion at that higher level, then the resource

handler moves to the yet next, higher level and makes the necessary conversions through two levels to provide the requested resource block.

Similarly, if a resource block is returned to the resource pool 12, the resource handler 11 returns the resource block, with its status changed from occupied to free, to the same position in the resource array as from where it was allocated. Moreover, the resource handler 11 also endeavors, in a preferred example embodiment, to return resources in a manner such that the same resources are not always allocated, e.g., a round robin selection and return. Instead, utilization of resource blocks at each level is substantially equally distributed over time amongst all of the resource blocks at that level.

Because the allocation and restoration of resource blocks to the resource pool hierarchy typically depends on external factors, the resource handler 11 preferably periodically, or when needed, reconfigures the resource pool 12 to attain a more efficient configuration. For example, a resource block currently allocated may be de-allocated and another resource block at that same level in a more efficient position in the hierarchy may be reallocated to that request thereby making available higher level resource blocks for future allocation.

Fig. 4 shows a resource hierarchy configuration example in which resources are efficiently allocated using the present invention in contrast to the inefficient, fragmented resource block allocation shown in Fig.2. Level 1 resource blocks are allocated in sequential fashion from left to right across each hierarchical level. In this example, level 1 resource blocks 20-23 are currently allocated. The next two level 1 resource blocks are allocated to a level 2 resource block 24. In response to a resource request for a level 1 or a level 2 resource block, the resource handler 11 would draw upon level 1 and level 2 resource blocks 25-27, respectively. As a result of this efficient resource allocation, a level 3 resource block 28, or alternatively, two additional level 2 resource blocks 29 and 30, is (are) available for higher level request allocations.

Fig. 5 shows a flowchart diagram illustrating example procedures (entitled Efficient Resource Allocation, at block 50) that may be followed by the resource

handler 11 for efficiently allocating resources from resource pool 12 in accordance with one preferred, but nevertheless example, embodiment of the present invention. The resource handler 11 analyzes an incoming resource request to determine the level of the request in the resource pool hierarchy or other structure (Block 52). For example, the resource request may require a certain number of resource units. Using the example hierarchical configuration shown in Fig. 1, if the request is for one resource unit, a level 1 resource block must be allocated; if the request is for two resource units, a level 2 resource block must be allocated; and if the request is for four resource units, then a level 3 resource block must be allocated, and so forth.

The resource pool configuration is mirrored in a list-based data structure stored in memory 11a accessed by resource handler 11. A corresponding list is maintained for each level in the resource pool configuration structure. A detailed example of such a list-based data structure will be described below in conjunction with the example embodiment of Figs. 8 and 9.

Based on that analysis, the resource handler 11 points to the corresponding level list (block 54) and decides if the corresponding list is empty (block 56). The list from level 1 (L1) is used as an example. In other words, the L1 list of resources in the resource pool 12 stored in the resource handler memory 11a is checked to detect any L1 resource blocks, i.e., from an L1 grouping with one or more resource blocks currently allocated.

If the L1 list is not empty, the resource handler 11 allocates the first available L1 resource block in the L1 list and removes it from the L1 list in memory 11a (block 58). On the other hand, if the L1 list is empty, the resource handler 11 determines whether the next level list, level 2 (L2) in this example, is empty (block 60). If there are entries detected in the L2 list, the resource handler 11 removes the first (if any) level 2 resource block from list L2 and adds a group of two (in this example) level 1 resource blocks to the L1 list (block 62). More generally, the resource handler 11 creates a group of level 1 resource blocks from a single level 2 resource block. One of the newly-added level 1 resource blocks is allocated by the resource handler 11 to the request and removed from the L1 list in resource handler memory 11a (block 64).

Alternatively, if there are no entries currently shown at list level L2 in memory 11a (block 60), the resource handler 11 determines whether the next list level, in this example the list for level 3 (L3) in the hierarchy, is empty (block 66). If not, the resource handler 11 selects the first available resource block from list L3 and converts that L3 resource block into a group of level 2 resource blocks (in this example two). This conversion effectively adds two L2 resource blocks to the L2 list and removes the L3 resource block from the L3 list (block 70). The resource handler 11 converts the first of the two, newly-created L2 resource blocks into a group of L1 resource blocks thereby adding two resource block entries to the L1 list (block 72). The converted L2 resource blocks accordingly removed from the L2 list in resource handler memory 11a. The resource handler 11 then allocates the first of the two newly-created L1 resource blocks to the resource request, and removes that L1 resource block from the L1 list in memory 11a. If the L3 list is empty, control proceeds to the next level (block 68), and similar procedures are performed.

Fig. 6 illustrates an Efficient Return of Resources routine (block 80) implemented by the resource handler 11. Each returned resource block (RB) is added to its corresponding level list (block 82). If resource blocks are allocated at one end of a particular list in the hierarchy or other structure, then it is preferable to evenly distribute utilization of resource blocks by returning resource blocks to the opposite end of that list so that all of the other resources at that level will be used before the just-returned resource is used again. This technique for allocating and de-allocating resources circulates the resource blocks so that they are all used substantially uniformly.

Returning a resource block to the resource pool 12, the resource handler 11 scans consecutive resource blocks at the lowest level list (block 84). A decision is made (block 86) if the consecutive number of resource blocks in this list form a resource block group (block 86). If so, the resource handler 11 converts them into a higher level resource block and corrects the lists corresponding to both levels (block 88). A decision is made (block 90) if the top list level is reached. If not, the resource handler 11 points to the next list level in memory 11a (block 92) and control returns to block 84 to repeat the operations

indicated in blocks 84-90. Thus, resource blocks are returned to the list and are grouped in such a fashion so that the highest level resource blocks can be made available.

Consider the following illustrative example. If the resources to be allocated represent hardware equipment, for example, it is advantageous to spread the resource allocations over all the resources. In that way, all the hardware is periodically checked for functionality rather than only when there is a heavy load requiring allocation of most or all resource units. This may be accomplished using the method outlined above with respect to Fig. 6 above, by always allocating free resources at a front end of the list, and returning the resources to the back end. The following illustrates this:

L4→ 256:

8

16

24

L3→ 128:

L2→ 64:

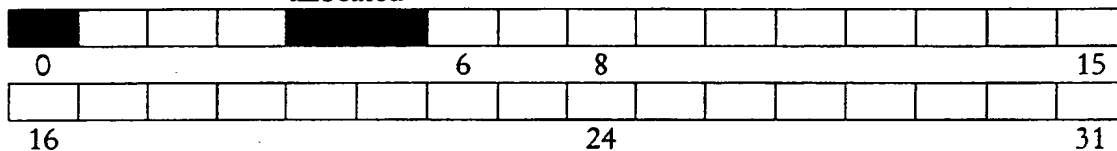
6

2

L1→ 32:

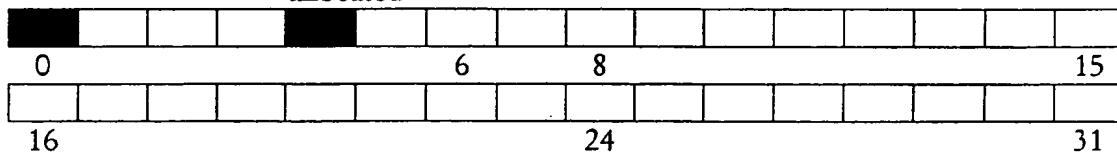
1

Currently
allocated



- 10 The resource at position 5 (one resource unit corresponding to capacity 32) is then returned to the last position in list L1. This leaves the following free resources:

Currently
allocated



The resource handler 11 also performs periodic resource management functions as will be explained in the example Resource Management routine (block 100) shown in Fig. 7. The resource handler 11 determines whether the resource pool

configuration is currently fragmented (block 102). If so, the resource handler 11 reconfigures the resource hierarchy or other structure (block 104) to de-fragment the current configuration. Certain resource requests may be reallocated to different resource blocks in order to group resource blocks so that the greatest number of highest level resource blocks are made available as described above.

The resource handler 11 also periodically receives statistics from statistics analyzer 13 continually gathered by statistics analyzer data processor 13a and stored in corresponding memory 13b before being transferred to the resource handler memory 11a (block 106). The resource handler 11 periodically analyzes the accumulated statistics (block 108) and decides, based on that analysis, whether a majority of the resource requests are occurring at a specific resource block level. For example, while the resource hierarchy is generally reconfigured in order to make available the resource blocks of the highest possible level, it may be that most of the resource requests occur at a level below the highest resource block level. Thus, in a simple example where there are three levels of resource blocks, it may be that only a small number of level 3 resource blocks are requested and that level 2 resource blocks are the most popular. Accordingly, rather than reconfiguring the resource pool which causes some disruption (albeit likely minimal disruption) in service for those allocated resource blocks that are reallocated, the resource handler 11 reconfigures the resource hierarchy to create as many level 2 resource blocks as possible (block 112). If a request is later made for a higher level 3 resource block, the resource handler 11 will generate (if possible) a level 3 resource block at the time of that request by combining (again, if available) a complete level 2 resource block grouping to create a single level 3 resource block. This type of statistical analysis and reconfiguration minimizes disruptions to service and also reduces data processing overhead required in combining and dividing various resource block groupings to make available a higher level resource block.

While the present invention may be advantageously applied to many environments, one example environment is portable radio communications. Fig. 8 illustrates in function block diagram format a portable radio communications system 150.

Mobile switching center (MSC) 156 interfaces a public switched telephone network (PSTN) 152 to the mobile communications network 150. A packet radio service node 158 interfaces data communications from the Internet 154 with the mobile radio communications system. Both the MSC 156 and packet radio service node 158 are coupled to a radio network controller (RNC) 160. Although only one RNC is shown for purposes of simplified illustration, those skilled in the art will appreciate that more than one RNC may be employed. The radio network controller 160 is coupled to a plurality of base stations (BS) 162. Portable radios illustrated as mobile stations (MS) 164 communicate over an air interface with one or more base stations 162 using a variety of protocols and procedures known to those skilled in the radio communications art.

Each base station includes (among other things) a resource handler 11 and resource pool 12 such as those shown in Fig. 3. Resource pool 12 is comprised of function blocks 68 where each function block is divided into various resource units that can be uniquely addressed and allocated to support various services provided by the portable radio communications system 150. The function blocks 68 in resource pool 12 can correspond to hardware and/or software resources required, for example, to assign radio channels to provide radio communications with the mobile stations 164.

Depending upon parameters of a particular call setup, more or less resources are required for the call. For instance, if the call is a basic voice communication, a smaller number of resource units is required than for a more complicated call, e.g., a multimedia call including voice, data, video, etc. Even among data calls there are different bandwidth requirements for high speed and low speed data calls with higher bandwidth data calls requiring more resource units than lower bandwidth data calls.

To accommodate wireless, multimedia communications, efficient resource unit allocation and return must be achieved so that multimedia and other high bandwidth calls requiring higher level resource blocks can be handled as well as lower level resource blocks associated with less demanding calls. A preferred access methodology that accommodates multimedia and other high demand communications services is wide-band code division multiple access (WCDMA). Calls are allocated across a very wide frequency

bandwidth, e.g., 5, 10, 15, etc. MHz, using orthogonal spreading codes, where one or more codes are assigned to each call. Each orthogonal spreading code is an example of a software-type resource block. Each base station is assigned a limited number of spreading codes.

5 Assuming that the mobile communications system 150 is a WCDMA system, the base station 162 may include a controller 170, an asynchronous transfer mode (ATM) transport 172, and a radio transceiver 174. An ATM transport interface to the radio network controller 160 is preferred because of its high efficiency and throughput. Controller 170 includes among other things a resource handler 11 and statistics analyzer 13
10 such as those shown in Fig. 3. The radio transceiver 174 includes a number of function resource blocks such as an encoder 176, a CDMA spreader 178, a modulator 180 for radio transmissions to the mobile stations 164. Additional function resource blocks are employed in the form of a demodulator 182, a CDMA de-spreader 184, and a decoder 186 for demodulating, de-spreading, and decoding radio transmissions from mobile
15 stations 164.

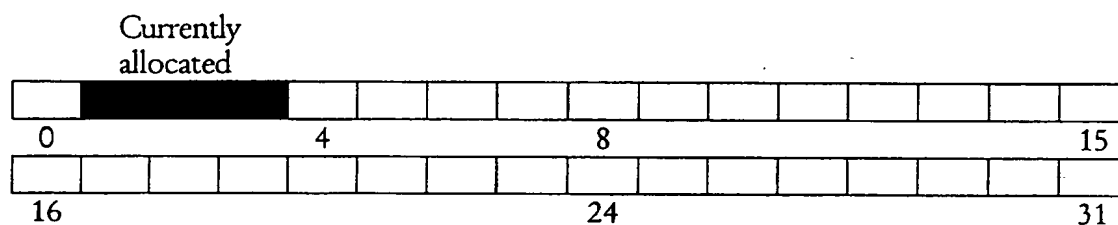
Each function resource block corresponds to one or more hardware and/or software resource units, where a resource unit is an integral functional unit having a unique address that can be individually addressed and operated. For example, an encoder employs both hardware and software resource units. When a resource unit is allocated to a resource
20 request, it is enabled and can then process data input provided by the requester.

The resource handler 11 maintains a "picture" of the resource units in a list data structure having a corresponding list for each free resource block level in its memory 11a. Each record/entry in a list represents one free resource block and may include information such as the type of resource unit (e.g., spreading code, encoder,
25 modulator, etc.), resource unit address, current configuration, current state (enabled/disabled), etc.

When setting up a path for a call through the base station 162, one or more resource units from the various different function blocks 176-186 must be allocated. For a

relatively simple voice call/path through the base station, an 8 kbps corresponding to a 32 kilosymbol per second (ksps) might use one encoder resource unit. The relationship between bit rate and symbol rate depends upon the particular coding employed in the encoder. On the other hand, a 64 kbps path might require two encoder resource units.

- 5 Consider an example where there are four resource block levels with level 1 corresponding to 32 kbps resource blocks, level 2 corresponding to 64 kbps resource blocks, level 3 corresponding to 128 kbps resource blocks, and level 4 corresponding to 256 kbps resource blocks. A list is maintained for each level L1-L4. As described above, if a list at a particular level is empty, the resource handler moves to the next level and
- 10 converts an available resource block into plural resource blocks at the next lower level. In this example, the base station 162 includes 32 resource units. A map of the free resource units at a particular time is shown below with shaded blocks corresponding to currently allocated resource units, each block is identified by an index with indices 0, 4, 8, 15, 16, 24, and 31 being shown.



- 15 Note that resource blocks 1, 2, and 3 are currently allocated. As mentioned above, one 32 kbps channel requires allocation of one resource unit or one of the 32 blocks. A 64 kbps channel requires two resource units and therefore must be assigned starting with index/address 0, 2, 4, ... 28 or 30. A 128 kbps channel requires four consecutive resource units starting with index positions 0, 4, 8, ... 24, or 28. A 256 kbps channel requires eight
- 20 consecutive resource units starting with index positions 0, 8, 16, or 24. Assuming that resource units are configured to provide the largest number of highest level groups as possible produces the resource structure shown below:

L4→ 256:

8

16

24

 L3→ 128:

4

 L2→ 64:
 L1→ 32:

0

Following the above example, a request for a 64 kbps channel means that the resource handler 11 scans the level 2 (L2) list. Since the L2 list is empty, the resource handler 11 converts a 128 kbps resource block into two 64 kbps resource blocks starting at index 4 which results in the following:

L4→ 256:

8

16

24

 L3→ 128:
 L2→ 64:

4

6

 L1→ 32:

0

5

The conversion provides two 64 kbps blocks at indexes 4 and 6. The 64 kbps block at index 4 is allocated (i.e., because it is first in the L2 list) and removed from the L2 list resulting in the following:

L4→ 256:

8

16

24

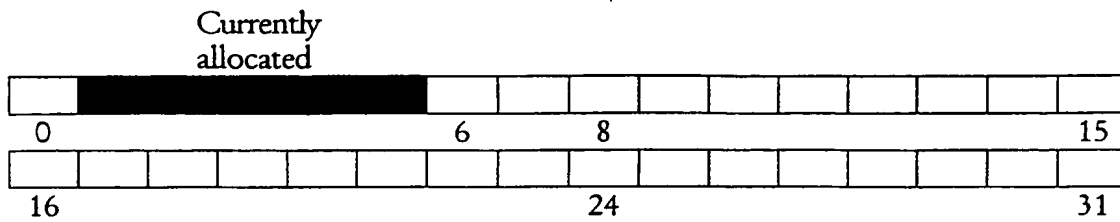
 L3→ 128:
 L2→ 64:

6

 L1→ 32:

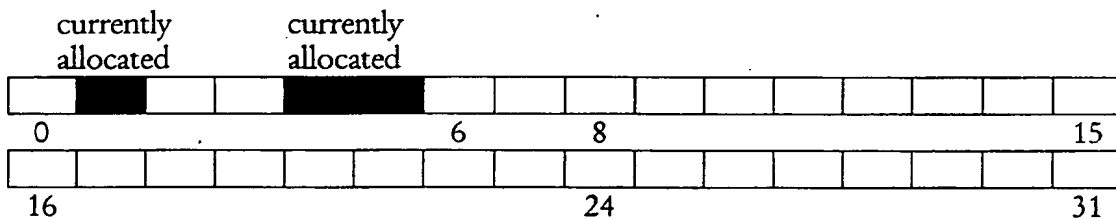
0

The current resource map is then as follows:



When resource units are returned (de-allocated), the resource units are placed last in the appropriate level list, i.e., the list having resource blocks of the same size. The resource manager 11 scans the list for consecutive resource blocks, and when the requisite number of consecutive resource blocks is detected, a next level resource block is created.

Consider for example the return of a 32 kbps resource block to the list of free 32 kbps resource blocks. Also assuming that the groupings include two resource blocks, if two consecutive 32 kbps resource blocks are in the list, the resource handler 11 creates a single 64 kbps resource block and deletes the two 32 kbps resource blocks from their respective lists. Consider the following example map of free resource blocks at the base station 162 having as mentioned above a total number of 32 resources.



Assume that the 32 kbps resource block at index position 1 is returned to the resource pool. The resource handler 11 enters that resource block last into the list of free 32 kbps resource blocks resulting in:

L4→	256:	<input type="text" value="8"/>	<input type="text" value="16"/>	<input type="text" value="24"/>
L3→	128:			
L2→	64:	<input type="text" value="6"/>	<input type="text" value="2"/>	
L1→	32:	<input type="text" value="0"/>	<input type="text" value="1"/>	

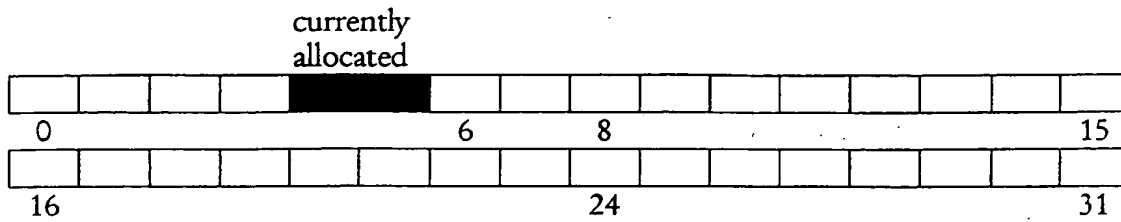
The resource handler 11 then scans this list for two consecutive resource blocks that can be grouped together to form a 64 kbps resource block as follows:

L4→	256:	<input type="text" value="8"/>	<input type="text" value="16"/>	<input type="text" value="24"/>
L3→	128:			
L2→	64:	<input type="text" value="6"/>	<input type="text" value="2"/>	<input type="text" value="0"/>
L1→	32:			

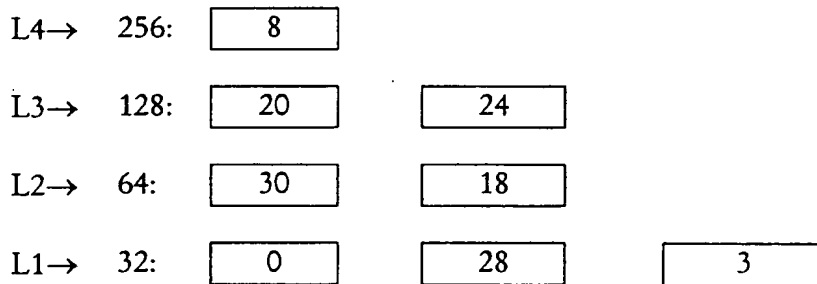
The resource handler 11 also scans for consecutive resource blocks in the 64 kbps resource block list. Since the two resource blocks start at zero and are consecutive, they are grouped together to form a 128 kbps resource block as follows:

L4→	256:	<input type="text" value="8"/>	<input type="text" value="16"/>	<input type="text" value="24"/>
L3→	128:	<input type="text" value="0"/>		
L2→	64:	<input type="text" value="6"/>		
L1→	32:			

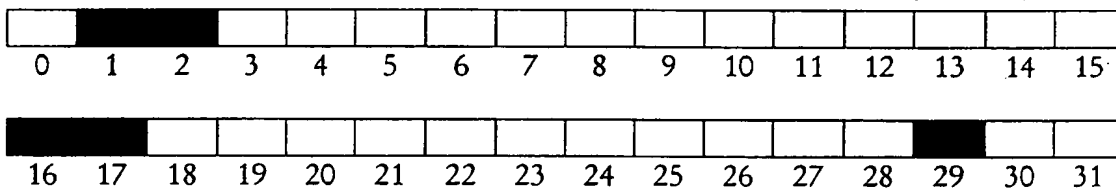
The resource handler 11 scans for consecutive resource blocks in the 128 kbps resource block list. Since there is only one resource block in this particular list, a 256 kbps resource block cannot be formed. The resource map after that allocation is as follows:



Such a list structure is also particularly advantageous in de-fragmenting the resource structure. The resource blocks to be moved are usually the resource blocks at lower level lists. Starting with the lowest level 32 kbps resource blocks, allocated resource blocks are moved to create as few larger level resource blocks as possible using the smallest number of de-allocations. Consider the following example of fragmented resources



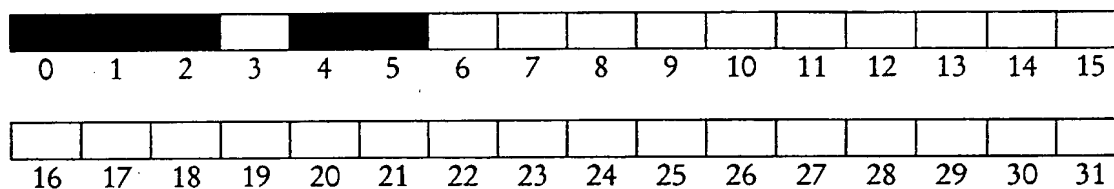
This configuration of fragmented resources is perhaps easier seen in the following resource structure map.



The resource structure may be more efficiently configured if the currently allocated 32 kbps resource blocks at index positions 1, 2, and 29 are allocated adjacent one another to make available larger resource blocks at their current respective positions. The first resource element at index position zero could be used to create a 64 kbps resource block if the 32 kbps allocation at index position 1 is moved to another index position. Accordingly, that 32 kbps allocation is moved logically to a buffer of allocated resources to

be relocated. The lists for each level are processed in the same fashion except the highest level at 256 ksps.

The resource units are reassembled minimizing the number of resource blocks that have to be moved. Each 256 ksps resource block is examined to determine how many resource blocks need to be moved to complete that 256 ksps resource block. In this example, only two blocks from positions 16 and 29 need to be moved to the first position (starting at index position 0) of the first 256 ksps blocks. After de-fragmentation, the 32 ksps resource blocks have been re-located to adjacent index positions 0, 1, and 2. The 64 ksps resource block now occupies index positions 4 and 5. The resource structure map appears as follows:



Another aspect of the present invention relates to efficient allocation of resources from different pools of resource units. A request for resources very often involves allocation of different types of resource units. For example, a resource request may require both software and hardware resources. In the context of a wideband CDMA radio communications system, a request for a resource corresponding to a communications channel requires allocation of a user-specific spreading code (a software resource unit) a signal processing and transceiver unit (a hardware resource unit). Each of the resource pools may be configured in a hierarchical tree, such as already described above, and each resource pool may have its own resource allocation strategy. Coordinating two or more resource allocation strategies for two pools of different types of resource units can be quite complicated and inefficient. Using different allocation strategies, the respective hierarchies fragment differently. As described above, a higher capacity parent resource unit can only be allocated if all of its lower capacity children are available. All of these factors may well produce situations where a resource unit of one type, e.g., a

spreading code, is allocated to a resource request but a corresponding and necessary resource unit of the other type, e.g., a transceiver unit, is not available (and vice versa).

The present invention solves these problems by establishing a relationship between resource units in the first and second resource pools. Although the example
5 embodiment employs two pools of different type resource units for description purposes, the invention applies to more than two resource pools. Referring now to the resource management system shown in Fig. 10, a resource handler 200 configures a first data structure 202 that mirrors the free resource units of a first type in a first resource pool included in the resource pools 210 using for example a hierarchical or other layered
10 structure. However, any structure could be used to organize the resources. A second data structure 204 is configured to mirror the free resource units of a second type found in a second pool of resources contained in resource pools 210 again preferably using a hierarchical or other layered structure. One example hierarchical structure is shown in Fig. 1 above including resource units at various levels 1, 2, 3, ... N. Resource units at each
15 level are configured into groups, each group linked to one resource unit at the next higher level. Thus, in the example in Fig. 1, two level 1 resource blocks form a level 1 resource unit group. A single level 2 resource unit is linked to one of the level 1 resource unit groupings. A single level 3 resource block is linked to a level 2 resource unit grouping corresponding in the example shown in Fig. 1 to two consecutive level 2 resource units.
20 An example of the first pool of resource units may be software resource units like CDMA spreading codes. The second pool of resource units may be hardware resource units such as signal processing and transceiving units.

The resource handler 200 includes a mapper 206 which establishes a relationship between ones of the resource units in the first pool mirrored by data
25 structure 202 and resource units in the second pool mirrored by data structure 204. Mapper 206 may establish various types of suitable relationship. For example, the relationship might include a one-to-one correspondence between individual resource units

in the first and second resource pools. In a hierarchical or layered configuration, the one-to-one correspondence could exist per unit per level.

A controller 208 controls allocation of resources in accordance with the relationship established by mapper 206 and also attempts to allocate lower level resource units so that higher level resource units are preserved for allocation. In other words, controller 208 first tries to allocate resource units from a grouping in which another resource unit has already been allocated. Only when all of the resource units in a partially-allocated grouping are currently allocated does the controller draw upon another resource group at that same level. When there are no available resource units at the requested resource level, the controller 208 converts an available resource unit from a next higher level (again from a group that is currently partially-allocated if possible) into a group at the requested resource unit level. The controller 208 also returns resource units to its resource pool and keeps track of which resource units are currently allocated and which are currently available. If possible, the controller 208 returns resource units in a manner so that the same resources are not always allocated so that utilization of resource units at each level is substantially equally distributed over time amongst all of the resource units at each level.

Referring now to the resource mapping routine (block 220) in Fig. 11, which may be employed by the resource handler 200, a first data structure is configured for a first pool of resource units, e.g., software resources like CDMA spreading codes (block 222). A second data structure is configured for a second pool of resources, e.g., hardware resources like radio transceivers (block 224). A relationship is then established between resource units in the first and second resource unit pools so that if one of the resource units from the first pool is available, a corresponding resource unit from the second pool is also automatically available (block 226). Because of this relationship, there only needs to be one resource unit allocation procedure to coordinate and efficiently allocate resource units from two (or more) resource pools that are required to satisfy a specific request (block 228).

Fig. 12 shows an example implementation of a mapper 206 as a matrix preferably implemented in software. The first data structure 202 is shown as a multi-level, hierarchical data structure of software resources like CDMA spreading codes. Data structure 204 is also shown as a multi-layer, hierarchical data structure of hardware resources like transceiver units. In this matrix implementation, the software resources correspond to rows of the matrix and the hardware resources correspond to columns. Fig. 12 shows a simple, example one-to-one correspondence in which each row resource unit is connected to (via an active intersection in the matrix structure) a single column resource unit. Active matrix intersections can be modified or reconfigured to accommodate changing needs, the addition of new resource units, or the removal of resource units. While a matrix is shown as an example mapping mechanism, those skilled in the art will appreciate that other mapping mechanisms may be employed.

While the invention has been described in connection with what is presently considered to be the most practical and preferred embodiment, it is to be understood that the invention is not limited to the disclosed embodiment, but on the contrary, is intended to cover various modifications and equivalent arrangements included within the spirit and scope of the appended claims.

WHAT IS CLAIMED IS:

1. A method comprising the steps of:

configuring a data structure corresponding to a pool of resource units including:

first resource blocks, each corresponding to one or more resource units,

5 provided in groups of first resource blocks and forming a first, lowest level of the resource unit pool, and

second resource blocks, each second resource block corresponding to one of the groups of the first resource blocks, provided in groups of second resource blocks and forming a second, higher level of the resource unit pool;

10 receiving a request for a resource unit; and

allocating an available first resource block from one of the first resource block groups in which one of the first resource blocks is already allocated.

2. The method in claim 1, further comprising:

15 if no first resource block is available from a first resource block grouping in which one of the first resource blocks is already allocated, allocating a first resource block from a first resource block grouping corresponding to one of the second resource blocks.

3. The method in claim 2, wherein the pool of resource units further includes:

20 third resource blocks, each third resource block corresponding to one of the groups of the second resource blocks, provided in groups of second resource blocks and forming a third higher level,

the method further comprising:

if no first resource block is available from a first or a second resource block grouping, allocating a first resource block from one of the third resource block groupings.

4. The method in claim 3, further comprising:

25 receiving a request from a second resource block, and

allocating an available second resource block from a second resource block grouping which one of the second resource blocks is already allocated.

5. The method in claim 4, further comprising:

if no second resource block is available from a second resource block grouping in which one of the second resource blocks is already allocated, allocating a second resource block from a second resource block grouping corresponding to one of the third resource blocks.

6. The method in claim 1, further comprising:

restoring a returned first resource block to a first resource block grouping in which one of the first resource blocks is already allocated.

7. The method in claim 1, further comprising:

allocating resource blocks from one part of the resource unit pool, and restoring returned resource blocks from another part of the resource unit pool.

8. The method in claim 1, further comprising:

reorganizing the resource unit pool by moving first resource blocks from one grouping to another grouping to create an additional second resource blocks.

9. The method in claim 8, wherein the reorganizing step is performed only when there is a request for a second resource block and no second resource block is currently available.

10. The method in claim 1, further comprising:

determining which level in the resource unit pool receives a greater number of requests, and

reorganizing the resource unit pool by moving resource blocks from one grouping to another grouping to create an additional higher level resource block up to the determined level.

11. A method comprising the steps of:

configuring a data structure corresponding to a pool of data processing resource units as a hierarchy including:

first resource blocks, each corresponding to one of individual resource units,

forming a first lowest level of the hierarchy, and
second resource blocks, each second resource block corresponding to a group of
the first resource blocks,
forming a second, higher level of the hierarchy;
5 receiving a request for a resource unit;
allocating a first resource block available at the first level of the hierarchy; and
if no first resource block is available at the first level of the hierarchy, converting
one of the second resource blocks at the second level of the hierarchy into plural first
resource blocks at the first level of the hierarchy, and allocating one of the converted first
10 resource blocks.

12. The method in claim 11, wherein the data structure corresponding to the
pool of resource units further includes:

third resource blocks, each third resource block corresponding to one group of the
second resource blocks,

15 forming a third, higher level of the hierarchy, the method further comprising:
if no first resource block is available at the first or the second levels of the
hierarchy,

converting a third resource block at the third level of the hierarchy into plural
second resource blocks at the second level of the hierarchy,

20 converting the converted second resource block at the second level of the hierarchy
into plural converted resource blocks at the first level of the hierarchy, and allocating one
of the converted first resource blocks.

13. The method in claim 11, further comprising:

restoring a returned first resource block first to a first resource block grouping at
25 the first level of the hierarchy in which one of the first resource blocks is already allocated,
if the returned first resource block completes a first resource block grouping in the
first level of the hierarchy, converting the completed, first resource block grouping to a
second resource block at the second level of the hierarchy.

14. A communications system, comprising:
a pool of communications resource units configured in a structure having plural levels including:

first resource blocks, each first resource block corresponding to an individual resource unit, provided in groups of first resource blocks and forming a first, lowest level, and

second resource blocks, each second resource block corresponding to a first grouping of the first resource blocks, forming a second level above the first level; and

a communications resource manager allocating various ones of the resource blocks in response to communication resource requests, removing allocated resource blocks, and periodically reorganizing by moving one or more lower level resource blocks from one grouping to another grouping to create an additional higher level resource block.

15. The communications system in claim 14, wherein the communications resource manager reorganizes the structure only when there is a request for a higher level resource and no higher level resource is currently available.

16. The communications system in claim 14, wherein the communications resource manager reorganizes using a minimal number of moves of lower level resource blocks from one grouping to another grouping to create an additional higher level resource block.

17. The communications system in claim 14, wherein the communications resource manager allocates available resource blocks from one part of the structure and restores returned communications resource blocks from another part of the structure.

18. The communications system in claim 14, wherein the pool of communications resource units includes:

third resource blocks, each third resource block corresponding to a second grouping of the second resource blocks, forming a third level above the second level, and

wherein the communications resource manager allocates resource blocks so that a greatest number of higher level resources are maintained for allocation.

19. The communications system in claim 14, wherein the pool of communications resource units includes:

third resource blocks, each third resource block corresponding to a second grouping of the second resource blocks, forming a third level above the second level, and
5 wherein the communications resource manager restores returned resource blocks to create a greatest number of available higher level resources.

20. A communications system, comprising:

a pool of communications resource units configured using a list data structure stored in a memory, the pool of communications resource units having plural levels in a
10 hierarchy including:

first resource blocks, each first resource block corresponding to an individual resource unit, provided in groups of first resource blocks and forming a first, lowest level of the hierarchy, where the list data structure includes a first list storing a corresponding data record for each available, first resource block, and

15 second resource blocks, each second resource block corresponding to a first grouping of the first resource blocks, forming a second level of the hierarchy above the first level, where the list data structure includes a second list storing a corresponding data record for each available, second resource block;

a communications resource manager allocating various ones of the resource blocks
20 in response to communication resource requests, removing allocated resource blocks from a corresponding list, and periodically reorganizing one or more of the lists by moving one or more lower level resource blocks from one list position to another list position.

21. The communications system in claim 20, wherein the communications resource manager moves one or more lower level resource blocks from one list position to
25 another list position.

22. The communication system according to claim 20, wherein the pool of communications resource units configured using the list data structure includes:

third resource blocks, each third resource block corresponding to a second grouping of the second resource blocks, forming a third level of the hierarchy above the

second level, where the list data structure includes a third list storing a corresponding data record for each available, third resource block.

23. The communications system in claim 20, wherein the communications resource manager reorganizes one or more of the lists using a minimal number of moves of lower level resource blocks from one grouping to another grouping to create an additional higher level resource block.

24. The communications system in claim 20, wherein the communications resource manager allocates available resource blocks from one part of a list and restores returned communications resource blocks from another part of that list.

25. The communications system in claim 20, wherein the communications resource manager allocates resource blocks so that a greatest number of higher level resources are maintained for allocation.

26. The communications system in claim 20, wherein the communications resource manager restores returned resource blocks to create a greatest number of available higher level resources.

27. A communications system, comprising:
a pool of communications resource units configured using a data structure having plural levels including resource blocks forming at plural levels in the structure, where different resource blocks are logically linked in the structure to form various groupings;
a communications resource manager analyzing a communications request and allocating one or more of the resource blocks; and
a request monitor monitoring the number of requests for communication resources at each level in the data structure and selecting a level that receives a greater number of requests,
wherein the communications resource manager reorganizes the data structure by moving resource blocks from one grouping to another grouping to create additional resource blocks at the selected level.

28. The communications system in claim 27, wherein the communications resource manager reorganizes the data structure using a minimal number of moves of resource blocks from one grouping to another grouping.

29. The communications system in claim 27, wherein the communications
5 resource manager creates a higher level resource above the selected level of the list when such a higher level resource is specifically requested.

30. A communications system, comprising:
a pool of communications resource units including:
first resource blocks forming a first, lowest level corresponding to one individual
10 resource unit;
second resource blocks forming a second higher level, each second resource block logically linked to a group of first resource blocks, and
a communications resource manager analyzing a communications request such that
if the request is for a first resource block, allocating to the communications request a first
15 resource block available at the first level, and if a first resource block is not available at the first level, converting an available second resource block at the second level into a group of available first resource blocks, and allocating one of the converted first resource blocks to the communications request.

31. The communications system of claim 30, further comprising:
20 third resource blocks forming a third higher level, each third resource block linked to a group of second resource blocks,
wherein if first resource is not available at the first and second levels, the communications resource manager converts an available third resource block at the third level into a group of newly-available second resource blocks, converts one of the newly-
25 available second resource blocks into a group of newly-available first resource blocks, and allocates one of the converted first resource blocks from the newly-available first resource blocks to the communications request.

32. The communications system in claim 30, wherein the communications system is a radio communications system and the communications resources include digital signal processors (DSP) and each DSP corresponds to one or more resource units.

5 33. The communications system in claim 30, wherein the communications system is a radio communications system and the communications resources include radio transmitter circuit boards, radio receiver boards, and signal processing boards, wherein each board corresponds to one or more resource units.

34. The communications system in claim 30, wherein the communications system is a code division multiple access (CDMA) radio communications system and the
10 pool of communication resource units includes CDMA spreading codes.

35. The communications system in claim 30, wherein the resource manager allocates resource blocks so that a greatest number of higher level resources are maintained available for allocation.

36. The communications system in claim 30, wherein a first resource block from
15 a corresponding group of first resource blocks is available if another one of the first resource blocks from the corresponding group of resource blocks is currently allocated.

37. The communications system in claim 30, wherein the resource blocks are configured in a hierarchy and the resource manager allocates resource blocks from one part of the hierarchy and restores returned resource blocks to another part of the
20 hierarchy.

38. A method comprising the steps of:
configuring a first data structure corresponding to a first pool of a first type of resource units;
configuring a second data structure corresponding to a second pool of a second
25 type of resource units;

establishing a relationship between ones of the first type of resource units in the first data structure and ones of the second type of resource units in the second data structure;

receiving a request for a resource that requires one or more resource units from the first and second pools of resource units; and

using the established relationship to allocate available resource units from the first and second pools of resource units to respond to the request.

39. The method in claim 38, wherein the relationship includes a one-to-one correspondence between individual ones of the first and second types of resource units.

40. The method in claim 39, wherein the first pool of resource units includes first, lowest level resource units of the first type provided in groups, and second, higher level resource units of the first type provided in groups, each corresponding to one of the groups of the first level resource blocks, and

wherein the second pool of resource units includes first, lowest level resource units of the second type provided in groups, and second, higher level resource units of the second type provided in groups, each corresponding to one of the groups of the first level resource units of the second type.

41. The method in claim 38, wherein the establishing step includes establishing a matrix where each row in the matrix represents a resource unit of the first type and where each column represents a resource unit of the second type so that selection of a row or a column corresponding to a resource unit of one type also selects a column or row, respectively, corresponding to a resource unit of another type.

42. The method in claim 41, further comprising:
configuring the matrix so that each row selects only one of the columns.

43. The method in claim 42, further comprising:
reconfiguring the matrix.

44. The method in claim 38, wherein the first type of resource units are software resource units and the second type of resource units are hardware resource units.

45. The method in claim 38 as applied to a code division multiple access (CDMA) communications system, wherein the first type of resource units are CDMA
5 spreading codes and the second type of resource units are data processing and transceiving hardware.

46. A method comprising:
storing a first structure for a first type of resource units;
storing a second structure for a second type of resource units; and
10 establishing a relationship between the first and second structures so that if one of the first type of resource units in the first structure is available for allocation, a corresponding second type of resource unit in the second structure is available for allocation.

47. The method in claim 46, wherein each of the first and second structures are
15 hierarchies having plural levels, where plural resource units on a lower level are grouped to a corresponding single resource unit on a higher level such that all of the lower level resource units in the group must be available before the corresponding single resource unit on the higher level may be allocated.

48. The method in claim 46, wherein the relationship includes a one-to-one
20 correspondence between individual ones of the first and second types of resource units.

49. The method in claim 46, wherein the establishing step includes establishing a matrix where each row in the matrix represents one or more different resource units of the first type and where each column represents one or more different resource units of the second type so that selection of a row or a column corresponding to a resource unit of one
25 type also selects a column or row, respectively, corresponding to a resource unit of another type.

50. Apparatus comprising:

means for storing a first data structure corresponding to a first pool of a first type of resource units and a second data structure corresponding to a second pool of a second type of resource units;

5 means for establishing a relationship between ones of the first type of resource units in the first data structure and ones of the second type of resource units in the second data structure; and

means, receiving a request for a resource that requires one or more resource units from the first and second pools of resource units, for using the established relationship to
10 allocate available resource units from the first and second pools of resource units to respond to the request.

51. The apparatus in claim 50, wherein the relationship ensures that if one of the first type of resource units in the first pool is available for allocation, a corresponding second type of resource unit in the second pool is available for allocation.

15 52. The apparatus in claim 50, wherein the relationship includes a one-to-one correspondence between individual ones of the first and second types of resource units.

53. The apparatus in claim 51, wherein the first pool of resource units includes first, lowest level resource units of the first type provided in groups, and second, higher level resource units of the first type provided in groups, each corresponding to one of the
20 groups of the first level resource blocks, and

wherein the second pool of resource units includes first, lowest level resource units of the second type provided in groups, and second, higher level resource units of the second type provided in groups, each corresponding to one of the groups of the first level resource units of the second type.

25 54. The apparatus in claim 50, wherein the establishing step includes establishing a matrix where each row in the matrix represents a resource unit of the first type and where each column represents a resource unit of the second type so that selection

of a row or a column corresponding to a resource unit of one type also selects a column or row, respectively, corresponding to a resource unit of another type.

55. The apparatus in claim 50, further comprising:
means for configuring the matrix so that each row selects only one of the columns.

5 56. The apparatus in claim 50, further comprising:
means for reconfiguring the matrix.

57. A communications system, comprising:
first and second pools of respective communications resource units, each pool
configured in a structure having plural levels;
10 a mapper that maps ones of the first pool of resource units to corresponding ones
of the second pool of resource units;
a communications resource manager, in response to communication resource
request, that allocates one or more of the resource units in the first pool, which also
allocates the corresponding one or more of the resource units in the second pool based on
15 the established mapping.

58. The communications system in claim 57, wherein the first pool of resource
units includes software resource units and the second pool of resource units includes
hardware resource units.

59. The communications system in claim 57, wherein the communications
20 system is a code division multiple access (CDMA) radio communications system, wherein
the first pool of resource units are CDMA spreading codes and the second pool of
resource units are data processing and transceiving hardware units.

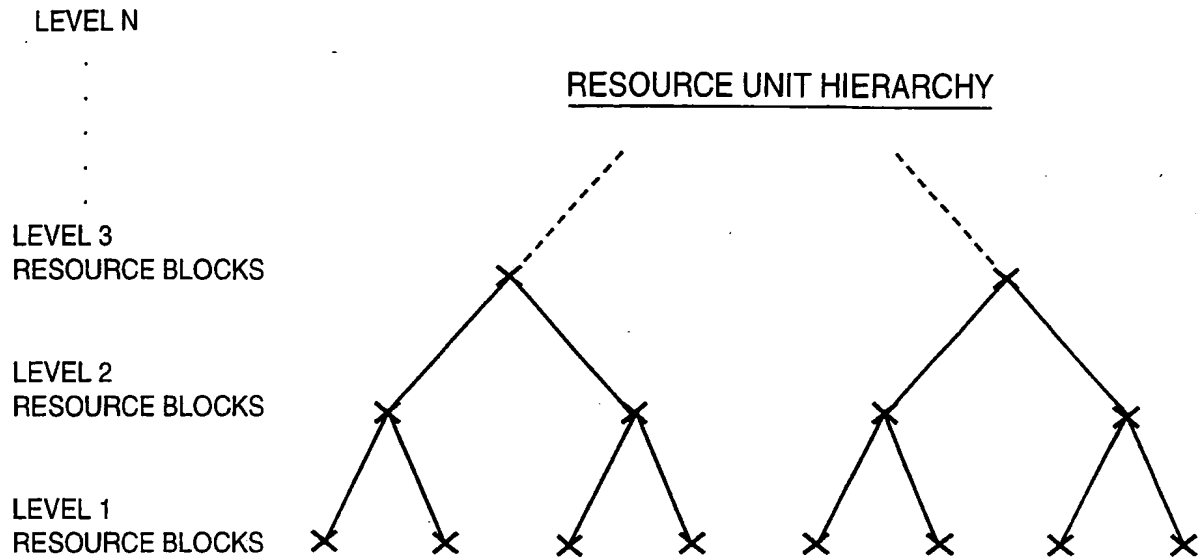
60. The communications system in claim 59, wherein the mapper includes
matrix where each row in the matrix represents a different CDMA spreading code and
25 where each column represents a different data processing and transceiving unit so that
selection of a CDMA spreading code or a data processing and transceiving unit also selects

a corresponding data processing and transceiving unit or CDMA spreading code, respectively.

61. The communications system in claim 62, wherein the mapper includes means for configuring the matrix so that each row is related to only one of the columns.

5 62. The communications system in claim 59, wherein the mapper includes means for reconfiguring the matrix.

1/10

**Fig. 1**

EACH LEVEL 1 RESOURCE BLOCK = 1 DATA PROCESSING HW OR SW RESOURCE UNIT

EACH LEVEL 2 RESOURCE BLOCK = 2 LEVEL 1 RESOURCE BLOCKS

EACH LEVEL 3 RESOURCE BLOCK = 2 LEVEL 2 RESOURCE BLOCKS

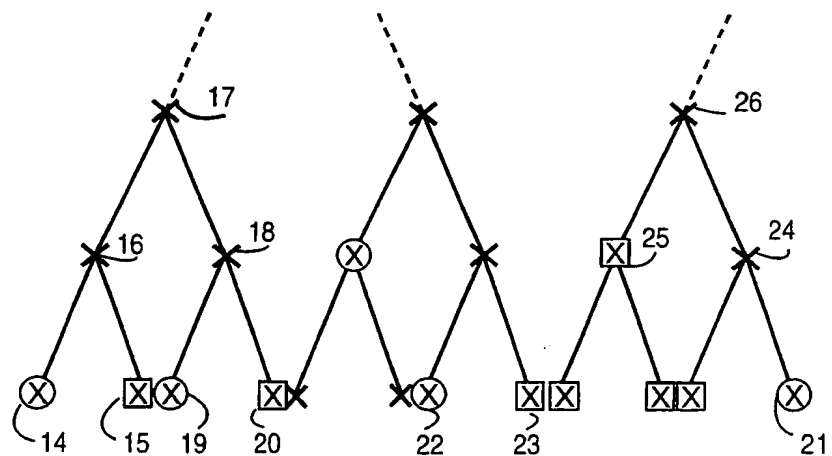
LEVEL N

...

LEVEL 3
RESOURCE BLOCKS

LEVEL 2
RESOURCE BLOCKS

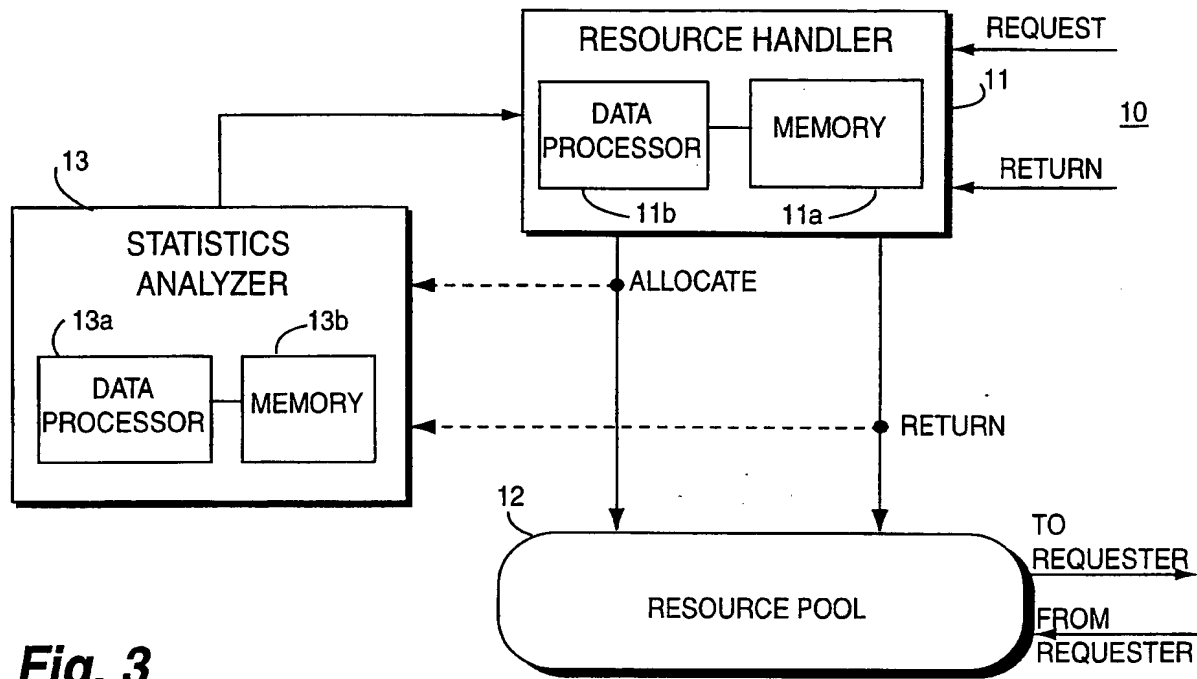
LEVEL 1
RESOURCE BLOCKS

FRAGMENTED RESOURCE ALLOCATION**Fig. 2**

⊗ = CURRENTLY ALLOCATED RESOURCE BLOCK

⊠ = AVAILABLE RESOURCE BLOCK

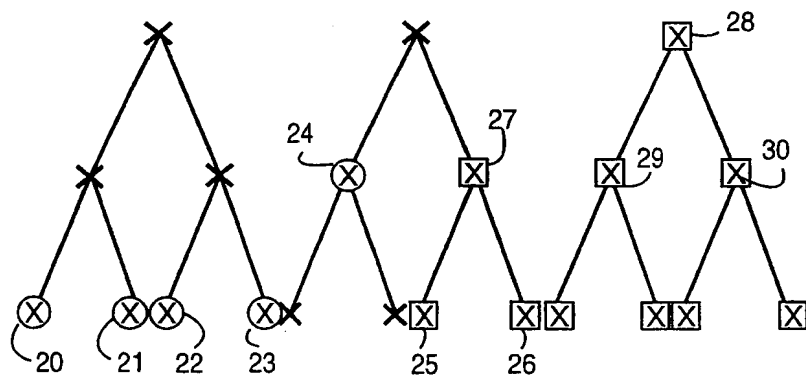
2/10

**Fig. 3**EFFICIENT RESOURCE ALLOCATION

LEVEL 3
RESOURCE BLOCKS

LEVEL 2
RESOURCE BLOCKS

LEVEL 1
RESOURCE BLOCKS

**Fig. 4**

⊗ = CURRENTLY ALLOCATED RESOURCE BLOCK

⊠ = AVAILABLE RESOURCE BLOCK

3/10

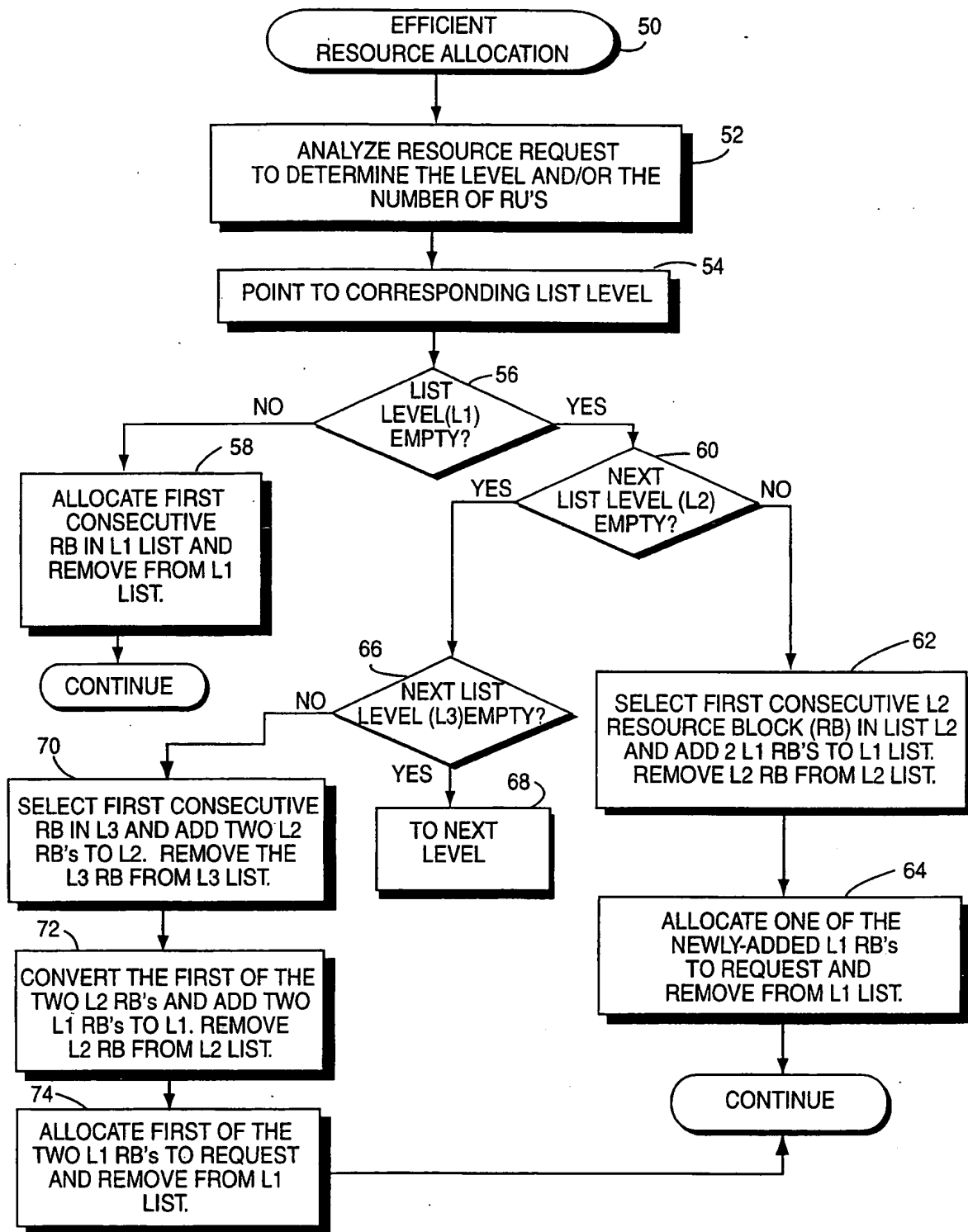
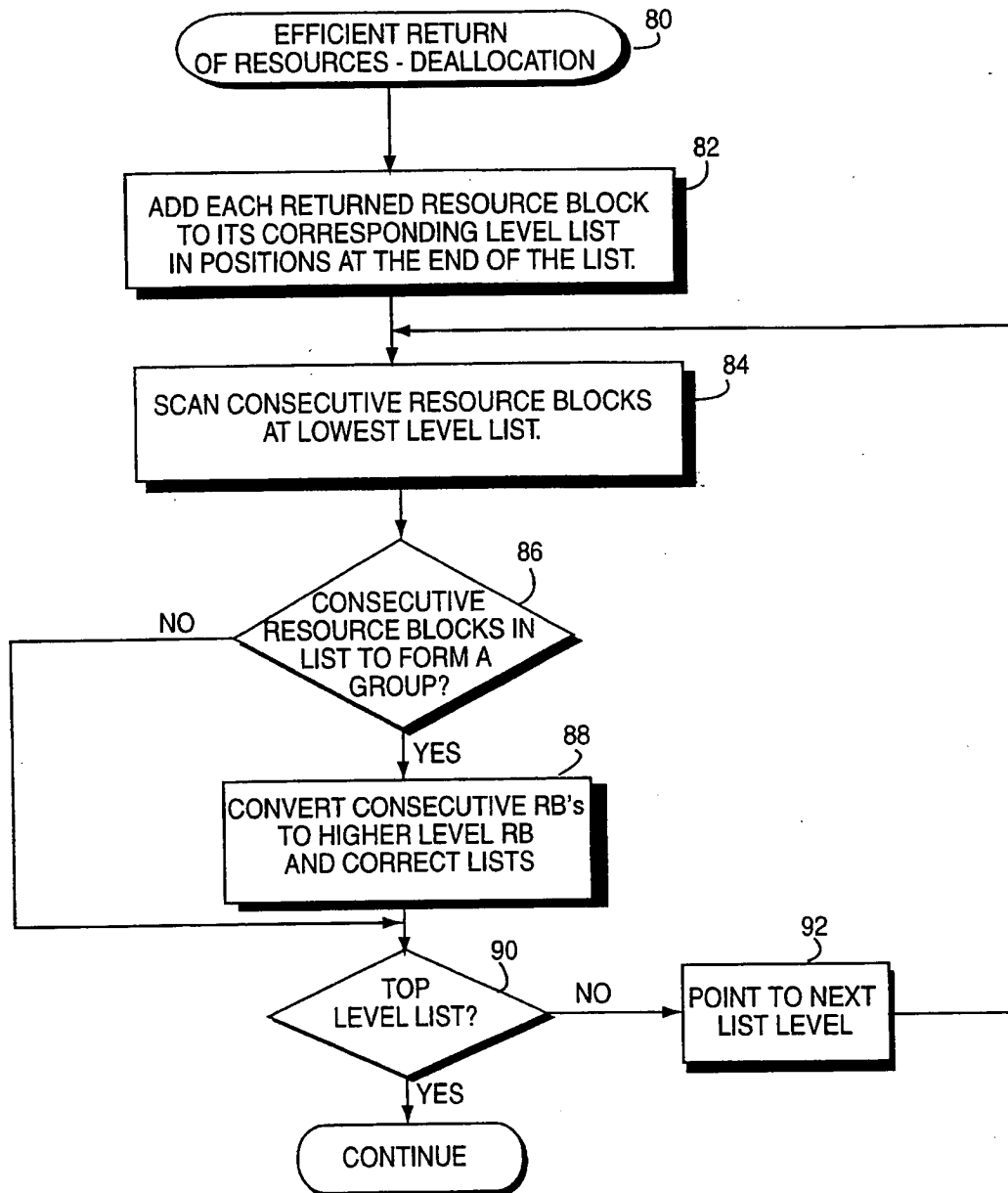
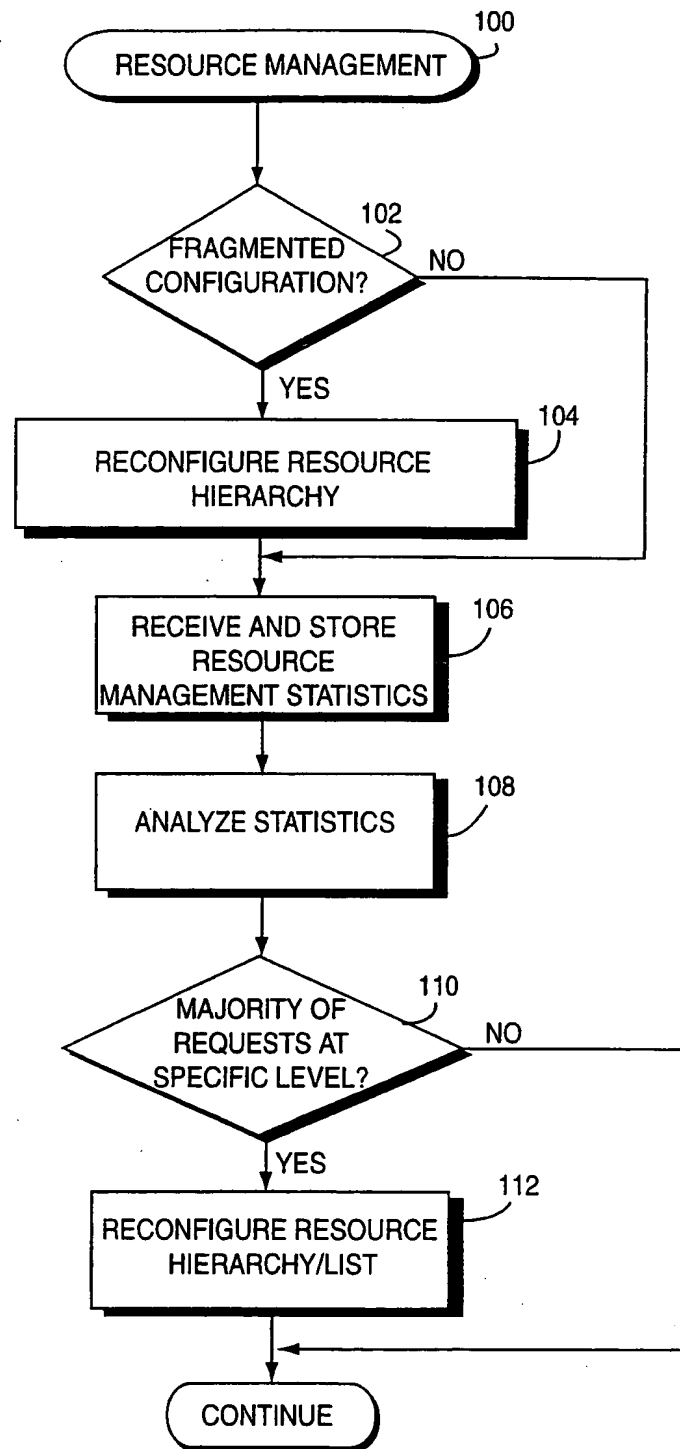


Fig. 5

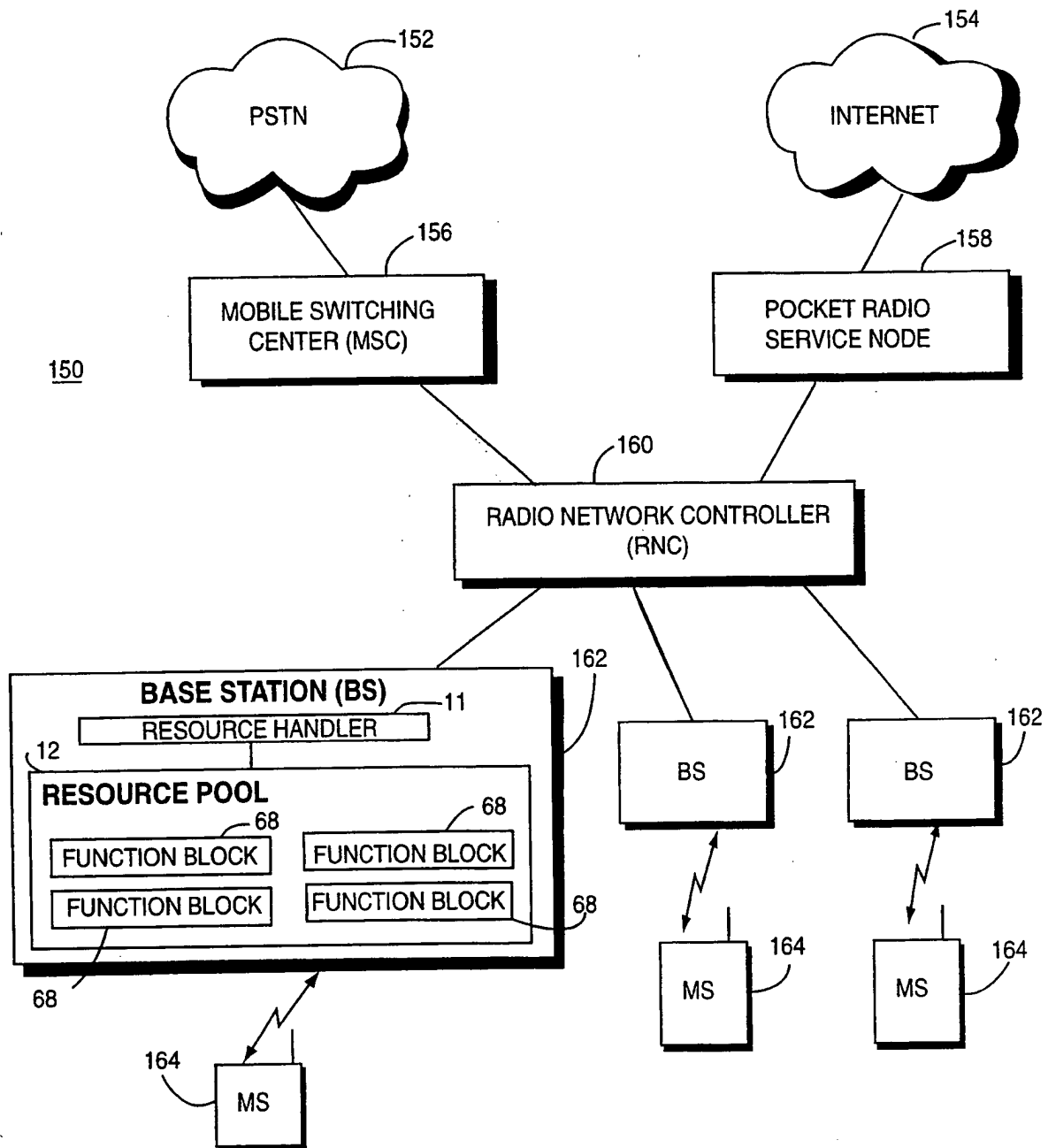
4/10

**Fig. 6**

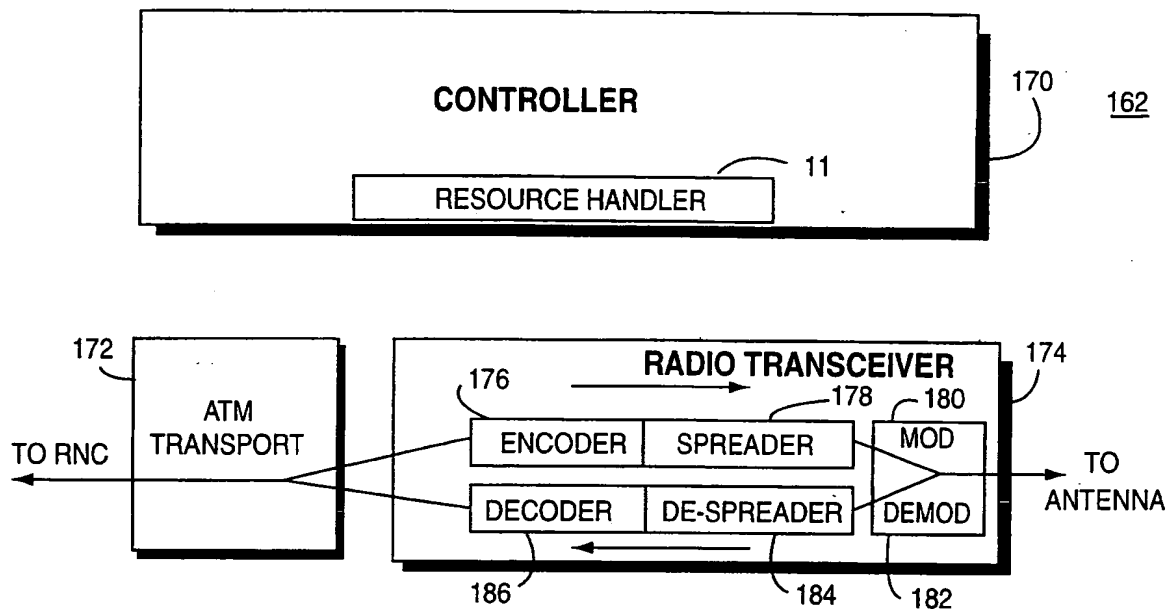
5/10

**Fig. 7**

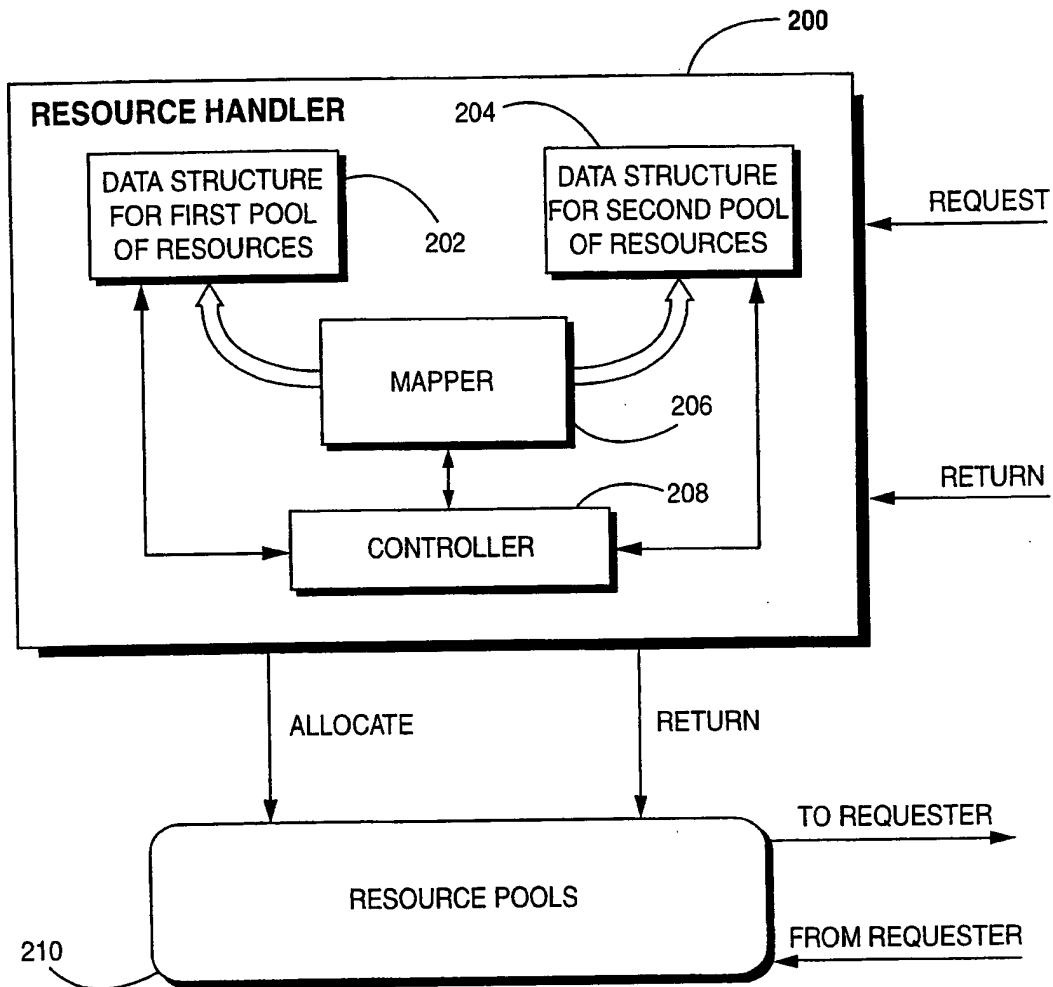
6/10

**Fig. 8**

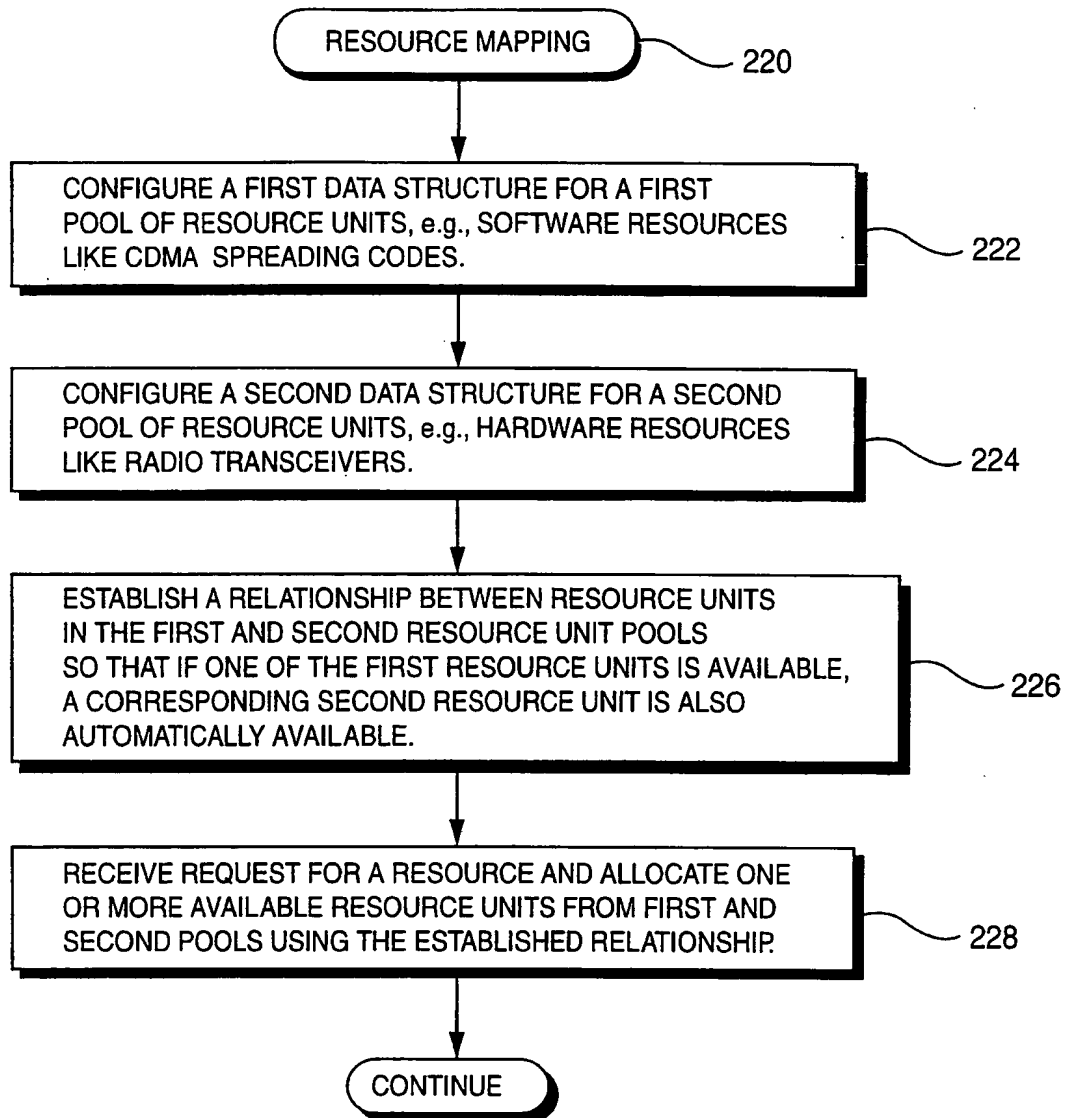
7/10

**Fig. 9**

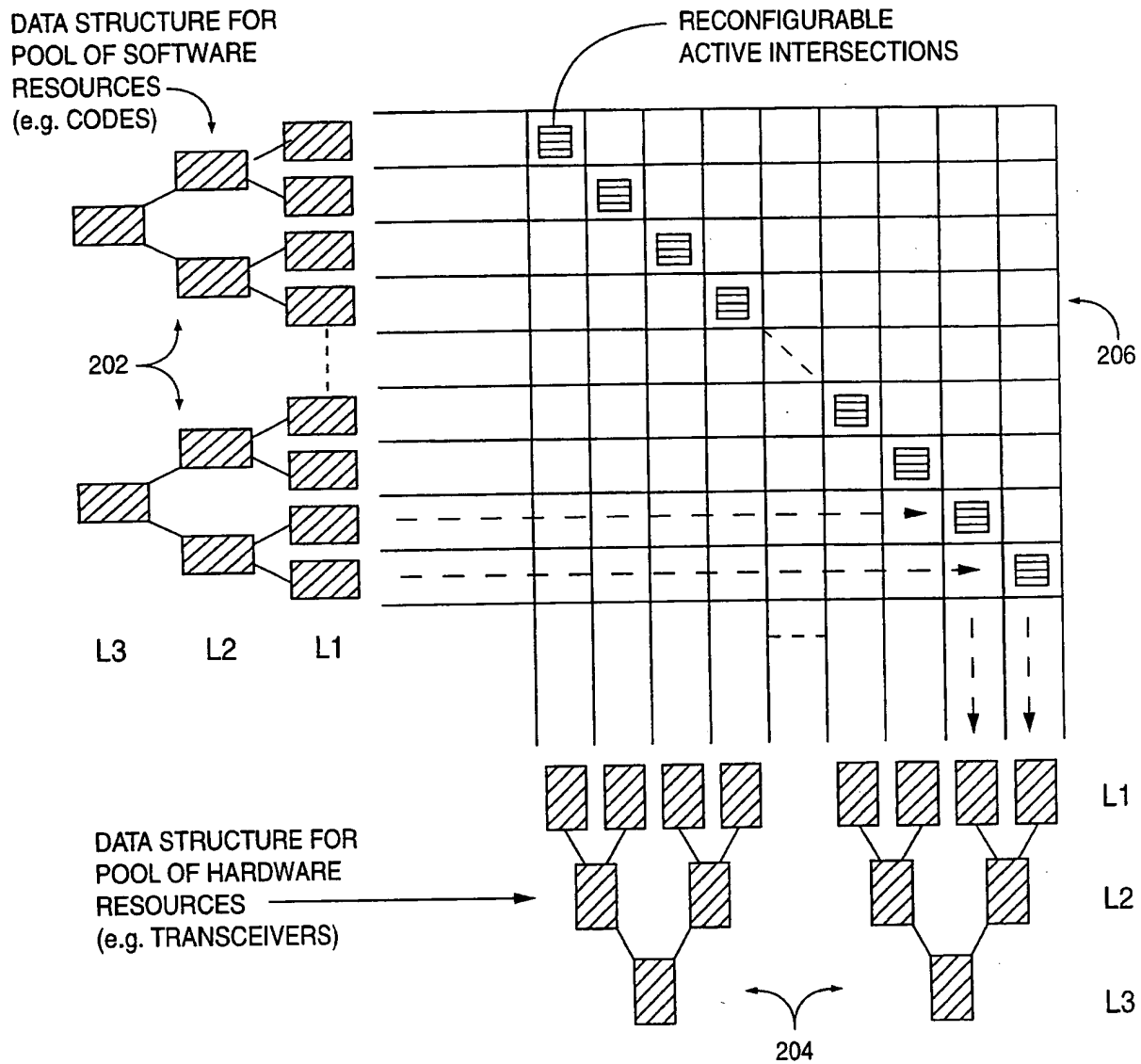
8/10

**Fig. 10**

9/10

**Fig. 11**

10/10

**Fig. 12**

INTERNATIONAL SEARCH REPORT

International Application No
PCT/SE 99/00664

A. CLASSIFICATION OF SUBJECT MATTER
IPC 6 H04Q3/00

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 6 H04Q

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	WO 97 22213 A (ANDERSSON STAFFAN ;ERICSSON TELEFON AB L M (SE); KONRADSSON LARS K) 19 June 1997 (1997-06-19) page 4, line 35 - page 7, line 9	1,14
A	figure 6 claims 1-3	2-13, 15-37
A	EP 0 658 062 A (AT & T CORP) 14 June 1995 (1995-06-14) the whole document	1-62
A	WO 95 34974 A (ERICSSON TELEFON AB L M) 21 December 1995 (1995-12-21) the whole document	1-37
	-/--	

☒ Further documents are listed in the continuation of box C.

☒ Patent family members are listed in annex.

* Special categories of cited documents:

- "A" document defining the general state of the art which is not considered to be of particular relevance
- "E" earlier document but published on or after the international filing date
- "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
- "O" document referring to an oral disclosure, use, exhibition or other means
- "P" document published prior to the international filing date but later than the priority date claimed

- "T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
- "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
- "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.
- "&" document member of the same patent family

Date of the actual completion of the international search

11 August 1999

Date of mailing of the international search report

20/08/1999

Name and mailing address of the ISA
European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040, Tx. 31 651 epo nl,
Fax: (+31-70) 340-3016

Authorized officer

Chassatte, R

INTERNATIONAL SEARCH REPORT

International Application No
PCT/SE 99/00664

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT		
Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 4 713 806 A (OBERLANDER LEWIS B ET AL) 15 December 1987 (1987-12-15) column 3, line 14 - column 8, line 31 column 39, line 1 - column 40, line 21 claims 1-17 ---	38-62
A	US 5 713 075 A (LIN SHIHCHAO ET AL) 27 January 1998 (1998-01-27) column 7, line 14 - column 9, line 56 ---	38-62
A	US 4 720 850 A (OBERLANDER LEWIS B ET AL) 19 January 1988 (1988-01-19) column 35, line 4 - column 39, line 13 ---	
A	WO 95 34975 A (ERICSSON TELEFON AB L M) 21 December 1995 (1995-12-21) the whole document -----	

INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No

PCT/SE 99/00664

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 9722213 A	19-06-1997	SE 508573 C AU 1048997 A EP 0865708 A SE 9504395 A	19-10-1998 03-07-1997 23-09-1998 09-06-1997
EP 0658062 A	14-06-1995	US 5440563 A CA 2123347 A,C CN 1110461 A JP 7177147 A	08-08-1995 13-04-1995 18-10-1995 14-07-1995
WO 9534974 A	21-12-1995	SE 502999 C AU 686827 B AU 2758895 A CA 2192791 A CN 1150875 A EP 0765555 A FI 964980 A JP 10504427 T SE 9500079 A US 5799153 A	11-03-1996 12-02-1998 05-01-1996 21-12-1995 28-05-1997 02-04-1997 05-02-1997 28-04-1996 14-12-1995 25-08-1998
US 4713806 A	15-12-1987	NONE	
US 5713075 A	27-01-1998	NONE	
US 4720850 A	19-01-1988	NONE	
WO 9534975 A	21-12-1995	SE 503021 C AU 2758995 A BR 9507994 A CA 2192792 A CN 1150874 A EP 0765556 A FI 965013 A JP 10501909 T NO 965272 A SE 9500078 A US 5696697 A	11-03-1996 05-01-1996 05-08-1997 21-12-1995 28-05-1997 02-04-1997 13-12-1996 17-02-1998 03-02-1996 14-12-1995 09-12-1997

THIS PAGE BLANK (USPTO)